

Application possibilities of open-source software for microbiological analyses

A. M. Danev^{1*}, R. N. Gabrova¹, T. Yaneva-Marinova², A. I. Angelov²

¹University of Food Technologies - Plovdiv, Dept. of Computer systems and technologies, 26 Maritza Boulevard, 4020 Plovdiv, Bulgaria

² University of Food Technologies - Plovdiv, Dept. of Biotechnology, 26 Maritza Boulevard, 4020 Plovdiv, Bulgaria

The main purpose of this paper is to present an approach for automatic microbiological analysis and in particular counting colonies using an open-source software for digital image processing. An open-source computer program called ImageJ is used for developing an algorithm for automatic detection and counting of lactic acid bacterial colonies grown in *MRS* agar. In comparison with commonly used techniques for manually counting of bacterial colonies and the expensive apparatus for microbiological analyses which provide this functionality, the algorithm presented in this paper gives an opportunity for fast, accurate and widely available technique for counting of bacterial colonies in agar medium. A comparative analysis between the proposed algorithm and well known *ImageJ* plugins for semi-automatic counting of bacterial colonies is made. The plugins that are tested are based on implemented in *ImageJ* methods for digital image processing. However, they do not provide a way for separating merged colonies. The results show that in comparison with the other *ImageJ* plugins with high average percentage (obtained after counting lactic acid bacterial colonies), the proposed algorithm achieves better accuracy with lower average percentage error.

Keywords: Open-source software, digital image processing, ImageJ, microbiological analysis, bacterial colonies counting

INTRODUCTION

Various microbiological analyses take a significant part in the food industry and modern technologies for food production and quality evaluation of raw materials. There are a different specialized apparatus and modern machines for performing microbiological analyses but they are too expensive and they can't be used freely by the wide audience. Such a highly specific and expensive apparatuses can be only used in special laboratories.

One of the most commonly used microbiological analysis is counting of bacterial colonies and determination the total microbial count in the sample. In order to perform such kind of analysis a sample with micro-organisms must be taken from the tested raw material or directly from a specific food. Then the sample is placed in a Petri dish called "Agar plate" filled with the culture medium for the specific microorganism. The agar plate can be used either to estimate the concentration of organisms in a liquid culture or the number of that culture in a suitably diluted sample. The micro-organisms contained in the sample begin to grow and multiply under the appropriate conditions forming genetically identical colonies. The colonies differ in shape, size and color depending on the specific microbiological culture. Different colonies can be formed on a single agar

plate. That is why it is important to recognize them and to know the exact number of all colonies grown in each agar plate.

The number of the formed colonies can be counted either manually or by using expensive apparatus with specialized application software for automatic bacterial recognition and using special freely available software applications for PC or smartphones. A brief overview for available free software products for microbiological analyses and in particular bacterial colonies counter applications is made in 2017 [1]. These freely available applications provide a fast and effortless way for enumeration of bacterial colonies, but the results are rather unsatisfactory in comparison to the traditional manual counting.

COMPUTER SOFTWARE FOR BACTERIAL COLONIES COUNTING USING DIGITAL IMAGE PROCESSING TECHNIQUES

Digital image processing is a technology widely used in computer science. Image processing takes a significant part in computer vision and pattern recognition. In brief, image processing using computer systems and applications is a method for performing some operations on an original image taken by a conventional camera, for example, in order to extract useful information needed for a specific analysis. There are various types of computer software applications for image processing as they differ in functionalities, accessibility and price.

* To whom all correspondence should be sent:
angel_danev_bg@abv.bg

Open-source software for digital image analyses – *imagej*

ImageJ is an open-source computer program for digital image processing which provides a lot of functionalities for processing ordinary images taken by a camera as well as scientific multidimensional images. The program is written in objective-oriented programming language *Java* and it is inspired by NIH Image which is a precursor of *ImageJ* [2 and 3]. Because *ImageJ* is written in *Java* the program is platform independent which can be run on any computer with *Java* 1.8 or later virtual machine installed. *ImageJ* has downloadable distributions available for Windows, Mac OS X and Linux. The program can work with the following file formats: TIFF, JPEG, PNG, GIF, BMP, DICOM, PGM, and FITS. Also other formats can be opened with the aid of special free plugins. Because of the fact that *ImageJ* is an open-source software, there are plenty of plugins, macros and extensions freely available for downloading and performing wide variety of tasks in different topics. Different useful programs and plugins for digital image processing based on *ImageJ* functionalities are developed in order to facilitate researchers working in the field of microbiological analyses [6, 11, 12, 13, 18, 19]. *Cell Counter* [15] is an *ImageJ* plugin for manual counting of bacterial colonies, based on simple interface. The plugin counts when the user clicks on the colonies in the image. There are plugins, such as: *Colony Counter* [16], *Colony Blob Count Tool* [17], *ColonyCountJ* [14] based on *ImageJ* functionalities for semi-automatically counting of cells and bacterial colonies.

ImageJ has a large user community which can be useful when a developer get involved with some problems and wants to implement new ideas. *ImageJ* is a powerful tool for fast image processing and the main purpose of the program is to serve on scientific community. A lot of scientific researches from different topics such as biomedical [3] and microbiological analyses, have been made based on the use of the functionality of *ImageJ*. In 2006 a new method based on image analysis using *ImageJ* for determining cyanobacterial biomass by CLSM in stratified benthic sediments are proposed [5]. In 2007 a different approach [7] of using *ImageJ* as an open-source software for digital image processing is presented. In [7] a quantitative analysis for histological staining and fluorescence using *ImageJ* is performed.

ImageJ has relatively easy to use and intuitive interface shown in Fig.1. After launching the program a main screen appears containing a well-

ordered menu bar and a tool bar with the commonly used tools for drawing and editing graphical objects.

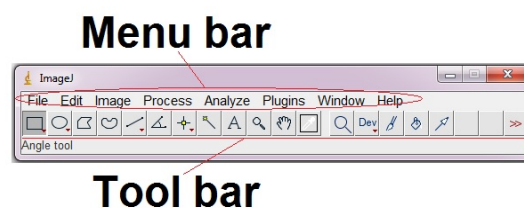


Fig.1. *ImageJ* graphic user interface

As an open-source software for digital image processing *ImageJ* provides various functionalities and plugins for different digital image transformations such as basic image editing tools; filtering algorithms like Mean, Median, Gaussian Blur, Minimum, Maximum, Variance filtering and thresholding on both 8-bit grayscale and RGB color images; tools for pattern recognition; possibilities for automating tasks and creating custom tools using macros; ability for implementing *ImageJ* class library in a different projects such as applets, servlets or applications where image processing is needed; tools for creating, editing and measuring tree types of selections such as rectangular, elliptical or irregular area selections; algorithms for image enhancement like edge detection, smoothing and sharpening; ability for adjustment brightness and contrast of the image; spatial operations such as crop, scale, rotate, resize, flip horizontally and vertically and so on. Some of the most significant functionalities of *ImageJ* are the various types of tools for image analyses such as areas measurements, standard deviation, mean, min and max of entire image or specific selection; tools for lengths and angles measurements using real world measurements units like millimeters. *ImageJ* gives the ability for different color processing in RGB and HSV color space and generating histograms.

Algorithm for automatic counting of bacterial colonies using *ImageJ*

Using some of the functionalities provided by *ImageJ* an algorithm for automatic counting of lactic acid bacterial colonies is developed. The workflow of the algorithm is shown on Fig.2. The algorithm is implemented in *ImageJ* as an additional macro.

For the proper operation of the algorithm a circular region of interest (ROI) is used. The diameter of the ROI is automatically pre-calculated according to the diameter of the agar plates that have been used. Once the ROI is set to appropriate

dimension and position a function provided by *ImageJ* is used for saving the ROI as a template and for future automatic using without the need to be calculated again.

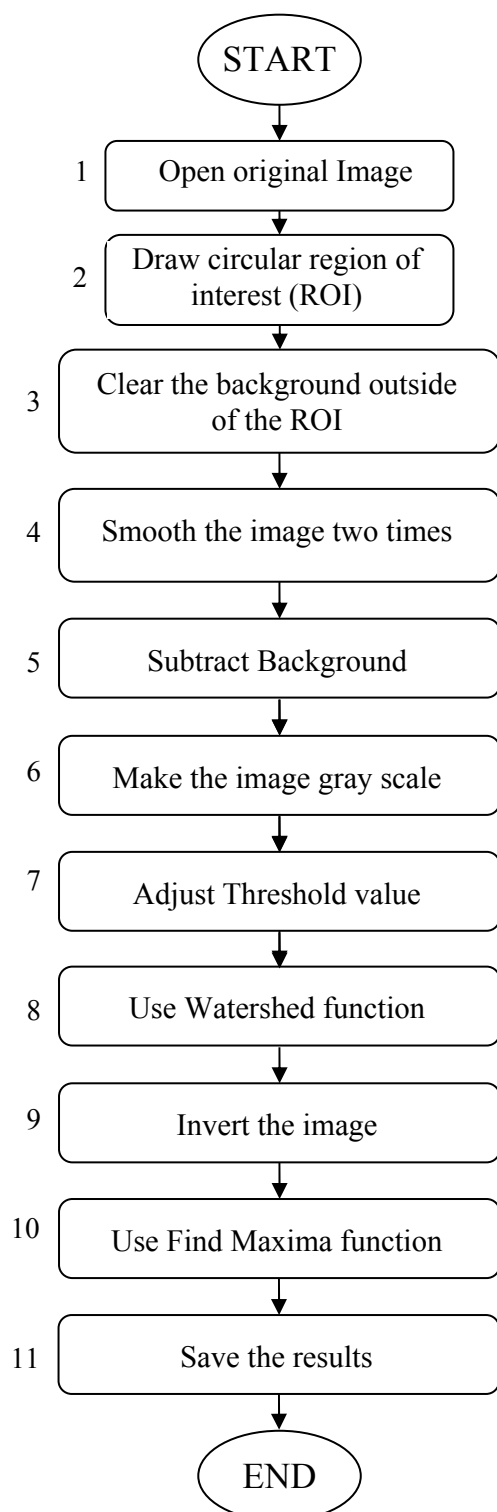


Fig.2. Workflow of the algorithm

During the analyses all agar plates are photographed using smartphone with 13mp camera under the same conditions such as: exactly the same

height between camera lens and the agar plate surface; all agar plates are positioned in the exact same spot on the background surface to ensure the exact positioning of the ROI used later.

The result from execution of some steps from the workflow diagram are shown on Fig.3. Some of the algorithm steps are not shown because the result can't be seen. After the analysis is complete, the results are saved in a text file from where they can be used to plot graphic or to perform some other operations on them.

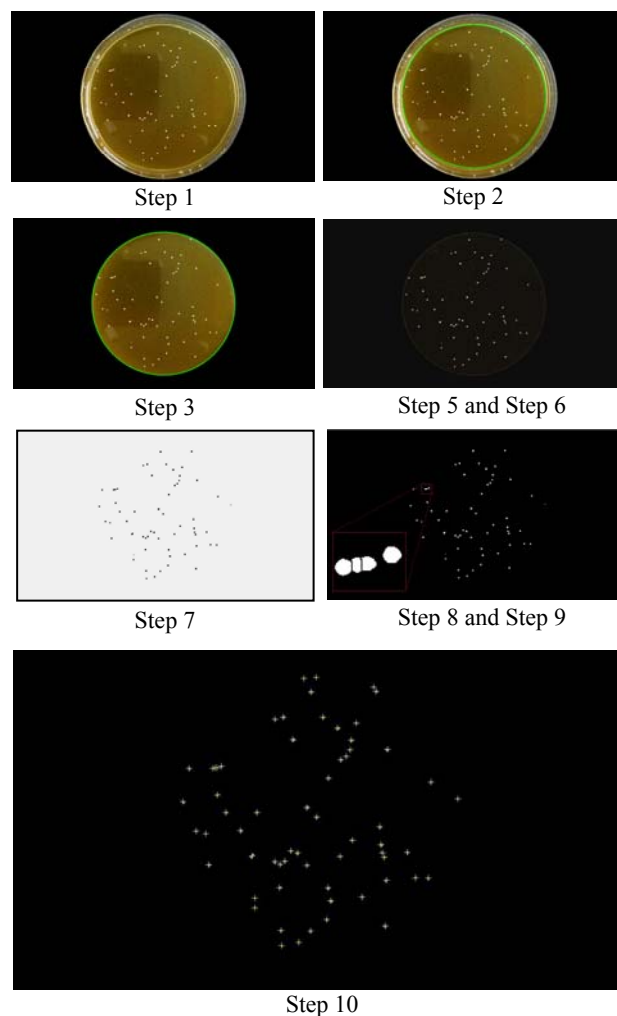


Fig.3. Result images after algorithm steps

Important steps from the algorithm presented above are adjusting the threshold value, using watershed function and using an *ImageJ* function called “Find Maxima”. *ImageJ* provides several popular algorithms for thresholding and for the purpose of this study a default thresholding algorithm is used which is a variation of the IsoData algorithm [8]. The required threshold value must be selected manually to achieve the best results. Using watershed function provided by *ImageJ* library is an essential step in the algorithm

for counting of bacterial colonies. The source code of the macro is shown on Fig. 4.

```
run("ROI Manager...");
roiManager("Open","D:\\BacterialColonies\\ROI\\ROI_1.roi");
roiManager("Select", 0);
setBackground(0, 0, 0);
run("Clear Outside");
run("Smooth");
run("Smooth");
run("Subtract Background...", "rolling=12");
run("8-bit");
setAutoThreshold("Default dark");
setOption("BlackBackground", false);
run("Convert to Mask");
run("Watershed");
run("Invert");
run("Find Maxima...", "noise=145
output=Count");
resultLabel = getResultLabel(nResults-1);
results table
count = getResult("Count");
str = resultLabel + ", " + count;
File.append(str, "D:\\Results\\Results.txt");
close();
```

Fig.4. Source code of the macro

Watershed segmentation is an algorithm for automatically separating particles touching each other. It is a common situation in the process of counting bacterial colonies to have several colonies overlapping each other. It is easy for the human eye to distinguish and count such overlapped group of colonies, but in terms of computer vision and object recognition things are different. This problem is shown on Fig.5. On the first image (on the left) the original image is shown where it is easy for the naked eye to differentiate tree overlapped colonies and one separated from the others. On the second image (the middle one) the same group colonies is shown but after thresholding the image. Here it is not so obvious how much colonies are overlapped. On the right image the overlapped colonies are separated using watershed function and it is easy for a computer program to precisely counting the exact number of overlapped colonies.



Fig.5. Using watershed function

Another important step in the algorithm for counting bacterial colonies is using the function called “Find Maxima” provided by *ImageJ* library. This function determines the local maxima in the selected region from the image, or in the entire image and then creates a binary image or one segmented particle per maximum, marked [9]. As an output type a *Point Selection* option is used which leads to tagging every object different from the background. In the example shown on Fig. 3 the objects representing the colonies are in white color and the background is in black color. As a result of using “Find Maxima” function, each object (colony) from the image is marked with “+” sign (Fig.3 Step 10).

RESULTS

The proposed algorithm for counting of bacterial colonies developed using the open-source software *ImageJ* provides a fast way for microbiological and biotechnological analyses. It doesn’t need special apparatus or any additional experimental settings for receiving the primary data, because using a smartphone camera for taking pictures is easy, quick and cheap way for getting the initial information from the objects under control.

For the purpose of this study photos of agar plates with colonies formed by lactic acid bacteria - *Lactobacillus plantarum* strain *BOM2* in an 8-hour fermentation process (under 37°C) of 6% oat suspension are analyzed. The bacteria have developed and formed colonies in the culture medium *Lactobacillus* MRS agar. Each of the photos is taken with a regular smartphone camera under a daylight. The total number of colonies in each agar plate can range from a few tens to few hundreds depending on the type of the dilution of the bacterial suspension prior to spreading them on the agar plates, and on the time for which the bacteria developed. According to the general rules for enumeration of bacterial colonies the total number of colonies on each agar plate must range from 30 to 300 [10]. For the purpose of this study and in order to determine the accuracy of the suggested algorithm, several agar plates with more than 300 colonies are included in the experimental results, but they can’t be used for performing accurate microbiological analyses. On the other hand, the colonies may differ in their size. As an example, some photos (nine) of the analyzed agar plates with different number of colonies and size are shown in Fig.6.

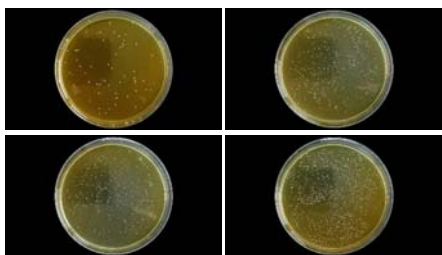


Fig.6. Agar plates with different total number of colonies

The total number of bacterial colonies for each of the analyzed agar plates is counted once manually and then automatically using a computer software with the algorithm described above. The analyzed agar plates, shown in Fig.7, are ordered according to the rate of dilution and the time, and the result data after manually and automatically counting is shown on Fig.8. For each group one, two or three agar plates (samples) with different total number of bacterial colonies are analyzed. The percentage error is calculated by Eq.(1) and it varies between 0.94% and 3.44%.

$$\text{Percent error} = \frac{N_{mc} - N_{ac}}{N_{mc}} * 100 \quad (1)$$

where:

N_{mc} - total number of colonies counted manually;

N_{ac} - total number of colonies counted automatically;

Tab.1 shows the average results for each group of samples which have been transformed in order to achieve appropriate results for plotting a growth curve.

The main purpose of counting the total number of bacterial colonies in agar plates is to plot the growth curve of microbial population. It is shown on Fig.9. The microbial growth curve is a graphical representation of changes in cell counts, biomass, or other suitable microbial growth indicator for a given microorganism in a suitable medium. The curve can also be used for determination of the growth phases which are – Lag phase, Exponential phase, stationary phase and phase of extinction.

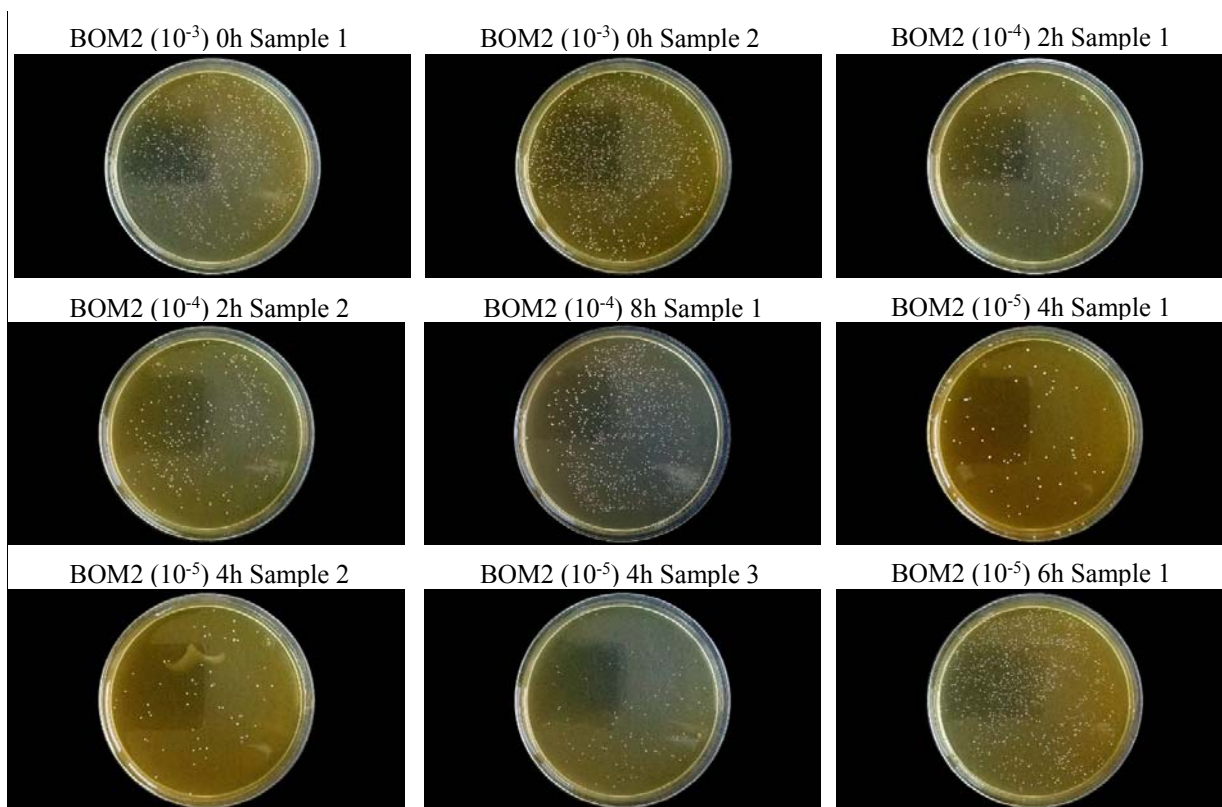


Fig.7. Analyzed agar plates ordered according to the rate of dilution and the time

Table 1. Results according to automatic counting of colonies formed by *Lactobacillus plantarum* strain BOM2 in an 8-hour fermentation process (under 37 °C) of 6% oat suspension

10 ⁷ CFU/ml	0h	2h	4h	6h	8h
<i>Lactobacillus plantarum</i> , BOM2	1,4000	2,8450	7,5666	11,6100	13,7500

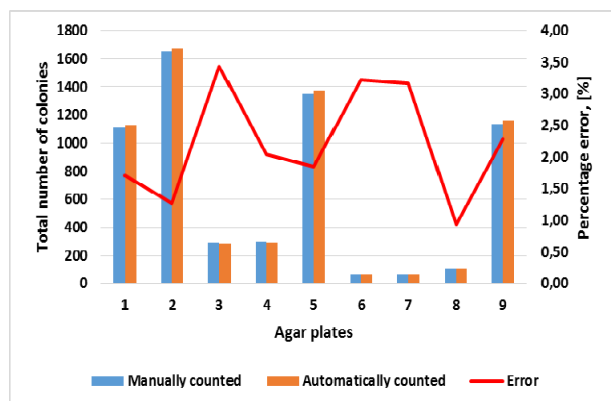


Fig.8. Graphic representation of results data and percent error

As it can be seen from Fig.8 the biggest percentage error is about 3.4% and the smallest percentage error is about 1%. The average error for this set of tested samples is about 2.2%. Therefore, the proposed algorithm can be successfully used for counting lactic acid bacterial colonies grown in *MRS* agar.

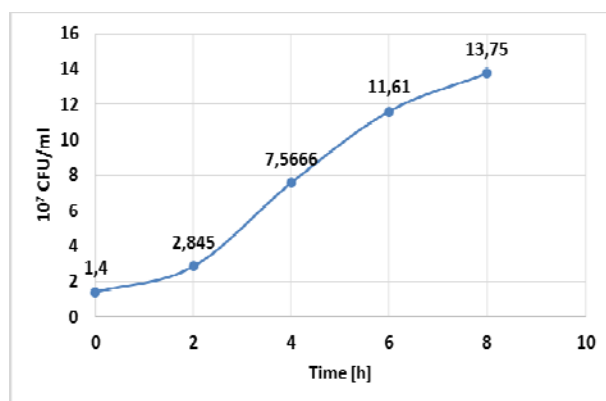


Fig.9. Growth curve of *Lactobacillus plantarum* BOM2 population

With respect to different *ImageJ* plugins and approaches for automatically counting of bacterial colonies, a comparative analysis of our own approach is made. The results achieved after using *Colony Counter* and *Colony Blob Count Tool* plugins are shown respectively in Tab.2 and Tab.3.

As it can be seen from Tab.2 the relative error (25.37) is very high and *Colony Counter* plugin can not be used for automatically counting lactic acid bacterial colonies strain *BOM2* grown in *MRS* agar. The plugin do not provide a functionalities for separating and counting merged colonies and for setting a region of interest.

The relative error after counting lactic acid bacterial colonies using *Colony Blob Count Tool* is 16.84%. This macro achieves better results than the plugin *Colony Counter*, but have some problems. To achieve these results the user have to pre-set

specific settings manually, depending on the sample being tested. Also, the user must know in advance the exact area (in pixels) of the smallest and the biggest colonies in the image. This leads to performing additional manual steps with *ImageJ* functionalities for measuring. This macro does not provide a functionalities for separating merged colonies. A result image after automatically counting lactic acid bacterial colonies using *Colony Blob Count Tool* is shown on Fig. 10.

Table 2. Results after counting with *Colony Counter*

Sample №	Manual Counting	ImageJ Colony Counter
BOM2 (10-3) 0h Sample 1	1108	865
BOM2 (10-3) 0h Sample 2	1653	929
BOM2 (10-4) 2h Sample 1	291	277
BOM2 (10-4) 2h Sample 2	294	286
BOM2 (10-5) 4h Sample 1	62	79
BOM2 (10-5) 4h Sample 2	63	83
BOM2 (10-5) 4h Sample 3	107	107
BOM2 (10-4) 8h Sample 1	1350	1180
BOM2 (10-5) 6h Sample 1	1135	1030
SUM	6063	4836
ΔA = Absolute Error [number of colonies]:	0	1227,00
Relative error [%]:		25,37

Table 3. Results after counting with *Colony Blob Count Tool*

Sample №	Manual Counting	ImageJ Colony Blob Count
BOM2 (10-3) 0h Sample 1	1108	870
BOM2 (10-3) 0h Sample 2	1653	1360
BOM2 (10-4) 2h Sample 1	291	270
BOM2 (10-4) 2h Sample 2	294	282
BOM2 (10-5) 4h Sample 1	62	54
BOM2 (10-5) 4h Sample 2	63	61
BOM2 (10-5) 4h Sample 3	107	110
BOM2 (10-4) 8h Sample 1	1350	1124
BOM2 (10-5) 6h Sample 1	1135	925
SUM	3302	2826
ΔA = Absolute Error [number of colonies]:	0	476,00
Relative error [%]:		16,84

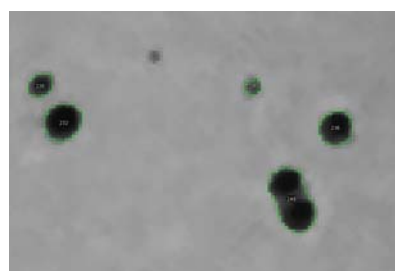


Fig.10. Result image after counting bacterial colonies with the macro *Colony Blob Count Tool*

The tested *ImageJ* plugins can be considered as programs for semi-automatic counting of bacterial colonies, because of the fact that the user have to perform additional steps using some of the *ImageJ* functionalities in order to evaluate the exact area of the colonies grown in the agar.

CONCLUSIONS

An algorithm for automatic counting of bacterial colonies in agar plates using freely available and open-source computer software for digital image processing is developed. The algorithm is based on build-in functionalities and tools in the program *ImageJ* and provides an approach for fast and accurate counting of bacterial colonies for the purpose of microbiological and biotechnological analyses. Several agar plates with different total amount of colonies formed by lactic acid bacteria *Lactobacillus plantarum* strain *BOM2* and *Lactobacillus MRS* agar medium are analyzed in order to evaluate the accuracy of proposed algorithm. The total number of colonies for each agar plate is counted in two ways – manually and automatically using algorithm implemented in open-source software as a macro. The results obtained from the manual counting and from the macro are compared and a percentage error is calculated. The obtained results and the analyses that have been performed lead to conclusion that the presented algorithm can be successfully used to determine the total number of bacterial colonies in the processes of microbiological and biotechnological analyses.

The algorithm mentioned in this paper for automatic counting of bacterial colonies can be implemented as a useful smartphone application in the future work.

REFERENCES

- [1] Danev A. M., R. Gabrova, „Open-source software products for microbiological analyses. A mini review“, Scientific works of University of Food Technologies, 64(1), (2017).
- [2] Schneider, C. A.; Rasband, W. S. & Eliceiri, K. W. "NIH Image to ImageJ: 25 years of image analysis". *Nature methods*, **9(7)**, 671-675, (2012), PMID 22930834.
- [3] Schindelin, J., Rueden, C. T. & Hiner, M. C. et al. (2015), "The ImageJ ecosystem: An open platform for biomedical image analysis", *Molecular Reproduction and Development*, ISSN: 1098-2795 (Online), PMID 26153368.
- [4] Silva A. F., G. CarvalhoEmail, A. Oehmen, M. Lousada-Ferreira, A. van Nieuwenhuijzen, M. A. M. Reis, M. T. B. Crespo, "Microbial population analysis of nutrient removal-related organisms in membrane bioreactors". *Applied Microbiology and Biotechnology* **93(5)**, 2171–2180 (2012).
- [5] A. Sole, J. Mas, I. Esteve, "A new method based on image analysis for determining cyanobacterial biomass by CLSM in stratified benthic sediments". *Ultramicrosc.* **107**, 669–673, (2007).
- [6] T. J. Collins, "ImageJ for microscopy". *BioTechniques* **43(1)**, 25-30 (2007).
- [7] Jensen E.C., "Quantitative analysis of histological staining and fluorescence using ImageJ". *The Anatomical Record* **296**, 378–381, (2013).
- [8] Ridler, TW & Calvard, S, "Picture thresholding using an iterative selection method". *IEEE Trans. on Systems, Man & Cyber.* **8**, 630-632, (1978).
- [9] Edelstein A., N. Amodaj, K. Hoover, R. Vale, Nico Stuurman. Computer control of microscopes using µManager. *Curr Protoc Mol Biol*, Chapter 14:Unit 14.20, 2010, doi:10.1002/0471142727.mb1420s92
- [10] Denkova Z. R., R. S. Denkova, V. B. Yanakieva, I. N. Dobrev, „Food microbiology. Exercise Guide“, UFT *Academic Publishing House of Plovdiv*, (in Bulgarian), (2015).
- [11] Masala G. L., U. Bottigli, A. Brunetti, M. Carpinelli, N. Diaz, P. L. Fiori, B. Golosio, P. Oliva and G. Stegel, "Automatic cell colony counting by region-growing approach", *IL NUOVO CIMENTO*, (2007), DOI 10.1393/ncc/i2007-10273-3.
- [12] Choudhry P. (2016) „High-Throughput Method for Automated Colony and Cell Counting by Digital Image Analysis Based on Edge Detection“. *PLoS ONE* 11(2): e0148469. doi:10.1371/journal.pone.0148469.
- [13] Ivan V. Grishagin, "Automatic cell counting with ImageJ", *Anal. Biochemistry* 473 (2015) 63–65.
- [14] Maurya D.K. (2017) "ColonyCountJ: A User-Friendly Image J Add-on Program for Quantification of Different Colony Parameters in Clonogenic Assay", *J ClinToxicol* 7: 358. doi:10.4172/2161-0495.1000358.
- [15] Kurt De Vos, University of Sheffield, Academic Neurology, [www] Available from: <https://imagej.nih.gov/ij/plugins/cell-counter.html>, Accessed on: 17.11.2018;
- [16] Vieira B., University of Lisbon, Colony Counter, [www] Available on: <https://imagej.nih.gov/ij/plugins/colony-counter.html>, [Accessed on: 06.12.2018].
- [17] ImageJ-macros, Colony Blob Count Tool, [www] Available on: http://dev.mri.cnr.fr/projects/imagej-macros/wiki/Colony_Blob_Count_Tool, [Accessed on: 06.12.2018].
- [18] Heras J., C. Domínguez, E. Mata, C. Larrea, V. Pascual C. Lozano, C. Torres, M. Zarazaga, GelJ – a tool for analyzing DNA fingerprint gel images, *BMC Bioinformatics* (2015), <https://doi.org/10.1186/s12859-015-0703-0>
- [19] Alonso CA, C. Domínguez, J. Heras, E. Mata, V. Pascual, C. Torres, M. Zarazaga, Antibigramj: A tool for analysing images from disk diffusion tests, *Comput Methods Programs Biomed.* (2017) 143:159-169, doi: 10.1016/j.cmpb.2017.03.010